Requirements Engineering:
Research Trends and Issues

by:
Abbas Rasoolzadegan
Reza Gorgan Mohammadi

Supervised by:
Ahmad Abdollahzadeh Barfroush

2011-11-30

Technical Report: CE/TR.C/90/02

No 424, Hafez Ave., Tehran 15914, Iran
**Abstract**

Requirements engineering (RE) plays a vital role in ensuring the overall success of the software engineering process. By virtue of such an importance, research in the field of RE has gained increasing attention. As RE research becomes diverse, the need for a quantitative analysis of the research activities using an engineering manner arises in order to provide researchers with a guideline to conduct their research in the RE field. Having reviewed the definitions for the terms “requirement” and “requirements engineering”, we have presented our own approach to define RE for engineering software in this paper. We have conducted an extensive meta-analysis of the RE literature related to our approach for the RE process. To do so, we have analyzed the trends and issues in several research resources using an iterative and evolutionary method. The results of our work are presented as an RE research tree. The paper familiarizes the RE researchers with our approach to RE and provides them with a quantitative analysis of the works performed on each activity in the process.

**Keywords:** requirements engineering process; new approach to definition; literature analysis; research trends and issues
1. Introduction

To have a successful business and increase company’s profit, it is necessary to develop products that satisfy the needs of the users. Discovering real user needs and transforming them into requirements is a challenging issue. Building software to facilities this issue is the main concern of a software engineer.

Requirements Engineering (RE), as a key process of software engineering, plays a crucial role throughout the software development lifecycle. Requirements engineering, a branch of software engineering, offers models, methods, metrics, measurements, indicators in a specified methodology to define and manage the needs of users systematically. Different studies [1,2] have shown that poorly defined software requirements, lack of user involvement, incomplete requirements and changing requirements are the major reasons why information technology projects do not deliver all of their planned functionality on schedule and within budget. Well-defined requirements will increase the likelihood of the overall success of the software project. RE lies at the heart of system development. In fact, RE bridges the gap between stakeholder goals and constraints, and their realization in systems. However, it will not be possible to develop better high quality requirements without a well-defined RE process.

Since requirements engineering is the starting point of software engineering and later stages of software development rely heavily on the quality of requirements, there is a good reason to pay close attention to the RE process. Over the past decades there has been increasing interest in requirements engineering and a large number of research papers have been published in the RE field. The motivation of this work is to get closer to the RE definition, activities, research trends and issues. This paper presents new approach to RE definition which mainly concern and put emphasis on transforming users wishes to software requirements. All related activities in this regard are introduced and research activities related to each topic are investigated using a meta-analysis of the RE literature. Given the aforesaid meta-analysis, the results allow a researcher to find the most important RE research for each topic. It is worth noting that we show our research results using various types of charts, figures, and tables, which finally result in the construction of an RE research tree.

The rest of the paper is organized as follows: in section 2 the motivation and background of the work is covered by describing why RE is important and what the state of RE definition is. Section 3, then presents a set of key terms that are necessary to communicate the scope and contributions of this work; An overview of requirements engineering including a brief history of RE emergence, a survey of significant definitions for the term “requirement” and “requirements engineering” as well as a quick review of the main phases of the RE process such as elicitation, analysis, specification, validation & verification, and management are given in section 3. Our understanding of the term RE and our suggestion for the RE process in this regard is given in Section 4. Section 5 presents related works. Section 6 describes the research method and limitations indicating how this study has been carried out. The quantitative and numerical results are illustrated by means of some diagrams and tables and finally by the RE research tree in section 7. Section 8 discuss future work and section 9 draws some conclusions.
2. Motivation and Background

Systematic transformation of user needs to software and managing different types of requirements have long been recognized to be a crucial phase of software development life cycle. There is an increasing awareness of the importance of RE. Numerous researchers have emphasized that the RE process is an essential contributor to the overall quality of the software product based on empirical investigations and industrial experiences [3, 4, 5, 6].

Moreover, a case study was carried out by Hall et al. in 12 companies at different levels of capability measured by using CMM [7]. The results show that out of a total of 268 development problems cited, almost 50% were requirements problems. All these studies suggest that RE plays a key role in software development.

Organizations that implement the effective RE process will enjoy multiple benefits including following:

- Clear definition of all types of requirements in general and particularly in large scale software development.
- Quantities of the quality of the system in determining the success or failure of project respect to quality, cost and efforts.
- Determining validation and verification of delivered system.
- Minimize the cost of maintenance phase in system life cycle.
- Improve cycle time productivity.
- Improve reengineering activities with preparing proper documents.
- Reducing rework during the later stages of development.

Effective RE process will provide quantitative W5H2 dimensions of software-to-be. The system-as-is has problems, deficiencies and limitations and the system-to-be is intended to address these problems based on business needs and technological opportunities.

It, the system-to-be, will be able to do so only if the organizational components, the users, and the technological components are able to cooperate. We need to figure out why the system-to-be is needed, what needs must be addressed and are vital to completion the project, who in this system will take part, when and where it should be on duty. It is also necessary to clearly define how it should be developed, operate and maintain, and how much cost, resource, labor, and effort it will take.

Requirements engineering is a process of discovering the needs of stakeholders and documenting them for analysis, communication and implementation. Many errors can be detected in the requirements phase. Fixing of errors detected in later stages of software development is more expensive than the initial stages. If errors are not detected on time, it leads to wrong product development. Wrong requirements can also lead to wastage of valuable resources.

Many of these errors are caused due to changes in the requirements. In order to eliminate errors there should be some measurement. Requirements metrics are important part of measuring software that is being developed. These include use case/size metrics, requirements traceability metrics, requirements completeness metrics, and requirements volatility metrics.
• **Use Case/Size Metrics**

Size is an important and general metrics used to measure requirements. Use Cases can be considered as a size measurement when they are used to describe functional requirements. Bullishness [8] discusses metrics for use case which can be derived by counting number of actors according to weights, number of use cases according to number of transactions that they perform, and number of analysis classes used to implement each use case according to their complexity.

An atomic action is used as a unit of measurement for use cases. Atomic actions cannot be further decomposed into deeper level. Bullishness [8] categorized use case metrics into 3 categories: 1) Use case size metrics includes number of atomic actions in main flow, number of atomic actions in alternative flow, longest path between first and last atomic actions of use case, and number of alternative flows. Size of use case can be determined by counting number of actors weighted by their complexity, number of use cases weighted by number of transactions they contain, and number of analysis classes used to implement each use case, 2) Use case environment metrics are the factors which have influence on complexity level of the use cases. These are independent of the size of the use cases. These include number of stakeholders, number of actors, and total number of goals, and 3) Use case composite metrics are derived from size metrics and environment metrics. They include total number of atomic actions in the alternative flows, total number of atomic actions in all flows, number of atomic actions per actor, number of atomic actions per goal, and number of goals per stakeholder.

• **Requirements Traceability Metrics**

Traceability is the ability to trace requirements in a specification to their origin from higher level to lower level requirements in a set of documented links. Traceability provides information which helps in determining whether all relationship and dependencies are addressed. This also makes requirements leakage - existence of lower level requirements with no valid origin from higher level visible. It helps in preventing wrong interpretations of other metrics. There exist 5 types of traceability metrics: 1) Next level coverage metrics (COB); this metric traces number of requirements to the next level up and next level down. This also traces number of requirements in both directions, 2) Full depth and height coverage (DECO; this is similar to COB but it traces requirements to highest and lowest level of specifications instead of only immediate next level in both directions, 3) Inconsistent traceability metrics; this include the numbers of requirements in a specification that have ablest one inconsistent link in both upward and downward directions, 4) Undefined traceability metrics; it includes the numbers of requirements in a specification that have no traceability links upward and download. However, the highest link will not have any upward link as it is derived from design document and the lowest level link has no down link, and 5) Linkage statistic metric; it measures the complexity level of traceability data by counting the number of higher or lower level requirements to which each requirement in a specification is traced.

• **Requirements Completeness Metrics**
Requirements Completeness Metrics are used to assess whether a requirement is at wrong level of hierarchy or too complex. A requirement document is said to be complete if it satisfies 4 conditions: 1) Everything the software is supposed to do must be included in the document, 2) All pages are numbered, all tables and figures have captions and numbers, all references are present, 3) No ‘TB’ (To Be Determined) should be present in the document, and 4) For every input of the system there must be some output.

This metric quantify decomposition level of higher level requirements allocated to a specification. These are marked with ‘to be completed’ tag and the number of it in lower level specification. Requirements Completeness Metrics are of 3 types: 1) Requirements Decomposition Metrics. This metric is used to know whether the requirements specified have been decomposed sufficiently to be used in the next phase. A complex function will have many levels of specification and a simple function will have few levels, 2) Requirements Specification Development. This metric is used to provide information about the work completed and the work remaining for a given specification. This metric is a solution to the question that queries have all the requirements in higher levels have sufficient low level specifications, and 3) Requirements Specification Metrics. This metric provide information about the quantity of items for which issue resolution is needed. It also provides information about the unresolved issues affected by higher level of requirements. This metric used ‘to be determined’, ‘to be supplied’ tags used for completing the requirements document. This also traces number of high level requirements allocated to specification for incomplete requirements.

- **Requirements Volatility Metrics**

The degree to which requirements change over a time period is called volatility of requirements. This metric is used to assess reasons for change of requirements over a period of time. Requirements degree of change is also checked by this metric. Both these factors are checked to know whether the changes are consistent with current development activities. It indicates changes such as addition, deletion and modifications. It helps in tracing future requirements, design and code volatility. Volatility can be high in the initial phase of software development. It should be reduced as the project progress so that further development should not be affected. This metric is a good indication of other metrics for example, if a specific portion of software is tested without knowing the volatility of the requirements than this testing would not be wrathful.

3. **Key Concepts and Definitions**

The terms *requirements* and *engineering* were first tied together by Alford in the development of SREM (Software Requirements Engineering Method) [9]. RE was initially applied to information systems, and hence was oriented towards organizational and application issues. Since the word *engineering* has been attached to *requirements*, RE research efforts has endeavored to incorporate an engineering approach to what was traditionally known as systems analysis.

RE traditionally was considered to be restricted to a particular phase of the software development life cycle which would normally occur before design, implementation, testing, and utilization. This view is based primarily on the waterfall model for
software development. However, this restrictive view of requirements has evolved a great deal in the last two decades. Some of the activities that were traditionally thought of as design, such as undertaking a feasibility study have become crucial to RE. Furthermore, it is now generally accepted that the requirements phase is not confined to the initial stage of the software development, as requirements are continually being refined throughout the life cycle.

During the last 2 decades, many various definitions have been proposed for the terms “Requirement” and “Requirements Engineering”. To make a common understanding about these terms, the definitions are presented and analyzed in this section.

3.1. Requirement

The concept of requirement is fundamental in requirements engineering. Generally speaking, the concept of “requirement” discussed in the RE domain refers to “what a system should do rather than how it should do it” [10]. The literature offers many definitions for the term requirement. Each definition emphasizes on some aspects and properties. Table 1 summarizes several definitions for the term sorted in ascending order by year.

<table>
<thead>
<tr>
<th>Index</th>
<th>Source</th>
<th>Definition</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abbott [11]</td>
<td>Any function, constraint, or other property that must be provided, met, or satisfied to fill the needs of the system’s intended user(s).</td>
<td>1986</td>
</tr>
<tr>
<td>2</td>
<td>IEEE [12]</td>
<td>(1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2).</td>
<td>1990</td>
</tr>
<tr>
<td>3</td>
<td>Davis [13]</td>
<td>A user need or a necessary feature, function, or attribute of a system that can be sensed from a position external to that system.</td>
<td>1993</td>
</tr>
<tr>
<td>4</td>
<td>IEEE [14]</td>
<td>A well-formed requirement is a statement of system functionality (a capability) that must be met or possessed by a system to satisfy a customer’s need or to achieve a customer’s objective, and that is qualified by measurable conditions and bounded by constraints</td>
<td>1998</td>
</tr>
<tr>
<td>5</td>
<td>SWEBOK [15]</td>
<td>A property that must be exhibited by a system developed or adapted to solve a particular problem</td>
<td>2000</td>
</tr>
<tr>
<td>6</td>
<td>Lethbridge [16]</td>
<td>A statement about the proposed system that all stakeholders agree must be made true in order for the User's problem to be adequately solved</td>
<td>2001</td>
</tr>
<tr>
<td>7</td>
<td>RUP [17]</td>
<td>A requirement is defined as &quot;a condition or capability to which a system must conform&quot;</td>
<td>2003</td>
</tr>
</tbody>
</table>
The requirements for a system are the descriptions of the services provided by the system and its operational constraints. These requirements reflect the needs of customers for a system that helps solve some problem such as controlling a device, placing an order or finding information.

Table 2 shows the topic/reference matrix for the definitions given in Table 1. The table is organized according to the keywords in the definitions of term “requirement”. In the rows, we have keywords of the definitions denoting the essence of a requirement and in the columns, we have the references. A “+” symbol indicates that the topic appears in the reference.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>What</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functionality</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraint</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capability</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Services</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statement</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Need</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In Table 3, in the rows, we have phrases denoting the desired goals of a requirement. As the table indicates, “solving a problem” is considered the most common goal for a requirement.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Solving a problem</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Satisfying a condition/constraint</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Satisfying a capability/property</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Achieving an objective</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.1.1. Types of requirements

The requirement definitions generally describe external and internal behaviour of system that must be met or provide user satisfaction on certain constrains and operational environment. The definitions emphasizes that a system must satisfy functional requirements which are recognized as a user need and system requirements which depends on non-functional, constrains and operational environment. According to the mentioned definitions, requirements can be categorized into product requirements and process requirements. The former contains functional and non-functional requirements; whereas the latter consists of operational constraints and properties of the operational environment. Figure 1 illustrates the classification.

Product requirements describe properties of a system or product. Process requirements describe activities performed by the developing organization. For instance, process requirements could specify specific methodologies to be followed and constraints that the organization must obey. Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement); a requirement for the product to be maintainable (a Product requirement) often is addressed by imposing requirements to follow particular development styles (e.g., object-oriented programming), style-guides, or a review/inspection process (process requirements).

A functional requirement specifies a function that a system or system component (i.e., software) must be capable of performing. Functional requirements describe what the system must do – an action that the product must take if it is to provide useful functionality for its users. Functional requirements are also called behavioural or operational requirements [13]. Functional requirements include input that the software gets and output it generates.

Non-functional requirements are those relating to performance, reliability, security, maintainability, availability, accuracy, error-handling, capacity, ability to be used by a specific class of users, acceptable level of training or support. According to Robertson
and Robertson [19], non-functional requirements are properties, or qualities, that the product must have such as appearance, speed or accuracy. Non-functional requirements describe how the system will do.

Differences between functional and non-functional requirements include, but are not limited to [20, 21]:

- Functional requirements usually relate to specific functions while nonfunctional requirements usually affect several functions (from a collection of functions to the whole system).
- Non-functional requirements are properties that the functions or system must have, implying that non-functional requirements are useless without functional requirements.
- When implemented, functional requirements either work or not while non-functional requirements often have a “sliding value scale” of good and bad.
- Non-functional requirements are often in conflict with each other, implying that trade-offs between these requirements must be made.

The correctness and quality of the system is going to be measured by functional and non-functional requirements. Furthermore, requirements cover not only the desired functionality and non-functional issues of a system or software product, but also address constraints and properties of operational environment on the design, implementation and run time.

Constraints are things that the products must include, conform to or be designed around. It should be noted that constraints are not necessarily negative; they are just things (usually components or systems) that the product must conform to. A simple way of looking at the difference between product requirements (functional and non-functional requirements) and constraints is to look at them as being essentially the same types of entities, but with requirements usually coming from the customer side, while constraints generally come from the company management side and from external regulatory bodies. Design and implementation constraints and properties are boundary conditions and assumptions on how the required software is to be constructed and implemented. Examples are including the system platform, budget, and the user ability to use the system.

3.1. 2. Levels of requirements

In addition to the different requirement types shown in Figure 1, an increasing number of sources point out that there are different levels of requirements as well (e.g. [21, 22, 23, 24, 25, 26]). Requirements can also be classified into three levels as: user wishes, business need, software requirements. Figure 2 illustrates our suggested classification for requirements levels.

![Figure 2. Levels of requirements](image-url)
User wishes refer to the user expectation of having a solution to achieve his/her optimal long term goal/s. Business needs refers to items of wish list which can be implemented based on technology availability. Software requirements are from need list which worth to be implemented respect to cost, usage and effort.

Sommerville [21] points out that some of the problems that arise during the RE process result from a failure to make a clear separation between different levels of requirements. Requirements have to be defined in all three levels within RE life cycle; Firstly, user wishes should be written using natural language and simple intuitive diagrams, because they are meant for people who do not have a detailed technical knowledge of the system. In second step, business needs, which are more detailed descriptions of the user wishes, are defined by gathering and analyzing the developers and stakeholders’ point of view based on cost, technical abilities and effort. High-level objectives of the organization and customer request from the system or product will be investigated in this stage. They analyze the feasibility of users’ wishes and system efficiency to refine the initial requirements from user. In third step, software requirements which are detailed descriptions of requirements for implementation will be defined. They form a model of the system that is usually too large and technical for users to understand. Models in UML language are examples of such a system model.

3. 2. Requirements Engineering

In recent years, various definitions elaborating the term RE form different points of view have been proposed. Table 4 summarizes the significant ones sorted in ascending order by year. Our proposed definition for RE is also given in the last row of this table.

<table>
<thead>
<tr>
<th>Index</th>
<th>Source</th>
<th>Definition</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ross [27]</td>
<td>Requirements engineering is a careful assessment of the needs that a system is to fulfill. It must say why a system is needed, based on current or foreseen conditions, which may be internal operations or an external market. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed.</td>
<td>1977</td>
</tr>
<tr>
<td>2</td>
<td>Loucopoulos [28]</td>
<td>Requirements engineering is a systematic process of developing requirements through an iterative, co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats and checking the accuracy of the understanding gained.</td>
<td>1995</td>
</tr>
<tr>
<td>3</td>
<td>Thayer &amp; Dorfman [29]</td>
<td>Requirement engineering provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification, and managing the requirements as they are transformed into an operational system.</td>
<td>1997</td>
</tr>
<tr>
<td>4</td>
<td>Zave [30]</td>
<td>“Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications”</td>
<td>1997</td>
</tr>
</tbody>
</table>
of software behavior, and to their evolution over time and across software families.”

“Requirements engineering covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a system. The term engineering implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent, relevant etc.”

Requirements engineering contains a set of activities for discovering, analyzing, documenting, validating, and maintaining a set of requirements for a system.

A systematic approach to eliciting, organizing, and documenting the requirements of the system, and establishing and maintaining agreement between the customer and the project team on the changing requirements of the system.

Appropriate mechanism for understanding what the customer wants, analyzing needs, negotiating a reasonable solution, validating the specification and managing changes in requirements.

The broad spectrum of tasks and techniques that lead to an understanding of requirements is called requirements engineering. From a software process perspective, requirements engineering is a major software engineering action that begins during the communication activity and continues into the modeling activity. It must be adapted to the needs of the process, the project, the product, and the people doing the work.

A systematic and disciplined approach to transform user wishes to software requirements based on business needs. Activities are included Eliciting, Analysis, Negotiation, Specification, Modeling, Validation & Verification, Identification, Change management, Traceability.

Table 5 summarizes the relations that we feel are between the key phrases in the definitions and the references. Similar to Table 2, a “+” indicates that the topic appears in the reference.
Table 5: Phases of Requirements Engineering

<table>
<thead>
<tr>
<th>Phase</th>
<th>+</th>
<th>+</th>
<th>+</th>
<th>+</th>
<th>+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negotiating a reasonable solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifying / Documenting requirements</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Validating the specification</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organizing requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Managing the requirements change (Maintaining requirements)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The use of systematic approach, techniques, and process</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

As table 5 indicates, analyzing, change management and systematic process are the most common phrases in the definition. Also some definitions reflects the nature of RE as a multidimensional discipline indicating that RE is not only related to technical issues and problems but also to managerial, organizational, economic, and social issues.

3.3. Requirements Engineering Process

As stated in Table 4, requirements engineering is a process itself. Numerous RE process models have been developed in the past two decades [10, 12, 34, 35, 36]. Each RE process model focuses on different aspects of RE and has different granularity. In this paper we propose a new approach for RE process, mentioned elaborately in section 4. In this approach the activities of the RE process are categorized into “Requirement Development” and “Requirements Management” activities. Requirements development activities consists of six main activities including requirements elicitation, negotiation, analysis, specification, modeling, requirements verification and validation (V&V). Requirements management activities include change management, traceability, and requirement identification. The mentioned activities are explained in the following sub-sections.

The final output of the RE process is the software requirements specification (SRS). A good software requirements specification should be unambiguous, complete, verifiable, consistent, modifiable, traceable, and usable [37].

3.4. Requirements Development Activities

- Requirements Elicitation

Requirements elicitation is a process of “identifying needs and bridging the disparities among the involved communities for the purpose of defining and distilling requirements to meet the constraints of these communities” [38]. Numerous techniques can be used in the requirements elicitation process [39]. Some examples are: interviews, questionnaires, ethnography, scenarios, use cases, and goal-based techniques [40, 41, 42, 43, 44].

- Requirements Negotiation

Negotiation is traditionally viewed as “the actual interactions among participants that lead to mutual commitment starting when participants begin communicating their goals and ending (successfully) when all agree to a specified contract.” [45]

Many existing methods neglect or do not explicitly address conflict handling and resolution. Nevertheless, negotiation techniques and tools have gained increased attention in software engineering research. Software engineering is a highly collaborative process and identifying shared or opposed interests is a necessity for project success [46, 47]. The objectives of customers, users, or developers have to be
understood and reconciled to develop mutually acceptable agreements [48]. This obviously does not mean that stakeholders will always agree. The result of negotiation is also to understand why stakeholders disagree. Identified disagreements represent major risks and need to be addressed by project management. Requirements negotiation is not a one-time episode in a project, but should be used early on and repeated in later stages [49]. In each cycle new stakeholders and new objectives have to be considered often leading to negotiations.

- **Requirements Analysis**

Requirements Analysis is a process during which requirements are assessed and analyzed, and their feasibility is examined. The elicitation process provides the input to this process. The output of the process is a consistent and complete set of requirements. Some typical techniques that can be used during this phase are: UML [50], Specification and Description Language (SDL) [51], Structured Analysis Structured Design (SASD) [52], Goal-based techniques [53, 54, 55], Petri Nets [56], and Conceptual Modeling Language (CML) [57].

- **Requirements Specification**

Requirements specification is the process of documenting the agreed requirements at an appropriate level of detail in the most suitable notation based on a well-defined document structure. This process has a very close relationship with requirements management. The key issues in the process are to select proper notations to specify requirements and at the appropriate level of detail. Additionally, the document structure and the labeling hierarchy of requirements are also important since they will directly relate to requirements traceability.

Entity Relationship Diagrams (ERD), Data Flow Diagrams (DFD), State Transition Diagrams (STD), Activity Diagrams, Use-Case Diagrams, Interaction Diagrams, and Formal notations are used for depicting requirements at various levels of detail.

- **Requirements verification and validation (V&V)**

Requirements verification and validation (V&V) is the process of examining the requirements specification to ensure that it is unambiguous, consistent and complete, and that the stakeholders are satisfied with the final requirements specification.

The major objectives of this phase are to:
- Verify requirements to ensure all the requirements are clearly defined, without ambiguity, complete, consistent, and follow the defined standard.
- Validate requirements with stakeholders to ensure the requirements are the real needs of stakeholders without having “not sure” requirements left in the requirements document.

The techniques used most often for this process are: Formal requirements inspection [58], Requirements testing, and Requirements checklist [59].

- **Requirements Modeling**

A model is the mathematical meaning of a description of a domain, or a prescription of requirements, or a specification of software, i.e., is the meaning of a specification of some universe of discourse [60].

Models in a narrow sense can be classified under various aspects. A model can mirror an existing original (like a photograph), or it can be used as a specification of
something to be created (like a construction plan). In the former case, we call it a
descriptive model; in the latter case, we call it prescriptive [61].

A descriptive model describes something already existing. A prescriptive model
models something as yet to be implemented. Thus domain specifications are
descriptive, while requirements specifications are prescriptive. A requirements
specification prescribes properties that the intended software shall satisfy. A software
specification prescribes certain kinds of computations [61].

Modeling is the act (or process) of identifying appropriate phenomena and concepts
and of choosing appropriate abstractions in order to construct a model (or a set of
models) which reflects appropriately on the universe of discourse being modeled [60].

3.5. Requirements Management Activities

Requirements management is the process of identifying, organizing, documenting
and tracking changing requirements in a project as well as the impact of these
changes. It is an ongoing task throughout the whole RE process and might span the
whole software lifecycle.

- Requirements Identification

Each requirement should be assigned a unique identifier so that it becomes
distinguishable. The identifier also facilitates requirements traceability.

- Requirements Change Management

Almost every software product continues to change and evolve throughout its
lifetime. Changes are inevitable because of system errors and better understanding
development of customers’ real needs. If change is not managed well, the quality of
the product will deteriorate and future changes will become increasingly difficult to
accommodate. Change management is mainly concerned with managing the current
status of all requirements, and placing requirements under configuration control [36].

- Requirements Tracing

There are several different definitions of the term requirements traceability (see e.g.
[62, 63, 64, 65]). We have chosen to define it as the “ability to describe and follow
the life of a requirement, in both forward and backward direction, ideally through the
whole system life cycle” [66].

As previously mentioned, every project is prone to changes during its development.
Changes to requirements cannot be totally avoided but they can be managed. Changes
can be due to addition of requirements. They can also be due to fixing of errors.
Changes can be done to meet new requirements or missed requirements in initial
drafts of requirements. Each proposed change should be evaluated on existing
requirements and system design and implementation should be modified. There could
be environment. Emergent changes due to customers new needs. Consequential
changes which are due to previous changes.

In order to maintain changes large database should be maintained. These changes
should be tracked on requirements document. One of the main tasks of requirements
management is to track the current status of the project. Database should be
maintained and updated daily about the current status. How many requirements are
covered and up to what level they are covered is known by status tracking. Stake
holders will be interested in knowing the current status of the product. Links should be
maintained to changed requirements. Traceability determines how the requirements
can be read, query and navigate in document. Requirements should be numbered dynamically. Database record identification and symbolic identification can also be used for numbering the requirements. Requirements must be traceable and traced to customers’ needs. These should also be traceable to work products like design and code.

4. RE Definition: New Approach

As indicated in Table 4, RE in terms of our definition is a systematic and disciplined approach to eliciting, organizing, documenting, analyzing, validating and managing changes in the requirements of the system. In this paper, the RE process is seen as a collection of well-defined activities, techniques, and transformations that people use to develop requirements of a system, and that includes a set of activities which implement certain functions. Based on the characteristics mentioned about RE process, we choose an idea inspired by the two-dimensional view of Rational Unified Process (RUP) [17] to illustrate and depict our RE process as shown in Figure 3.

The RE process is inherently complex and critical. The main reason is that the requirements engineering process has the most dominant impact on the capabilities of the resulting product. Furthermore, requirements engineering is the process in which the most diverse set of product demands from the most diverse set of stakeholders is being considered [67].

RE process is divided into two main groups of activities, requirements development and requirements management. Requirement development activities incorporates activities related to eliciting, analyzing, documenting, and validating requirements, whereas requirement management includes activities related to maintenance, namely identification, traceability, and change management of requirements.

In the authors’ view of requirements engineering, the process is iterative and incremental. RE activities cover the entire system and software development lifecycle and the process will go into more detail in each iteration. Traditional views of the requirements engineering process considers it just in the beginning of the system development lifecycle [68]. However, in large and complex systems development, developing an accurate set of requirements that would remain stable throughout the months or years of development has been realized to be impossible in practice [69].

Therefore, requirements engineering should be viewed as an incremental and iterative process, performed in parallel with other system development activities such as design. Accordingly, RE might not solely be a front end process, but also be part of the later stages of software engineering. In this paper, RE process consists of three main phases namely “Wish”, “Need”, and “Requirement”. Also, as it was mentioned in subsection D, the process consists of five main activities including elicitation, analysis, specification, verification and validation (V&V), and management. The first four are mainly related to technical issues and therefore are also called technical activities. Each phase can be considered as a process itself.

Based on the characteristics mentioned about RE process, we choose an idea inspired by the two-dimensional view of Rational Unified Process (RUP) [17] to illustrate and depict our RE process as shown in Figure 3.
The relationship between the three phases mentioned above is shown in Figure 4. The process commences by a primary set of user expectations. The output of the first phase is elaborated using a descriptive model. The outcome of the phase is what in the stage of Wish called “$\text{What}_W$” which can answer to questions like “why” and “who”. The second phase receives $\text{What}_W$ as input and elaborates it using a prescriptive model. The outcome of the phase is called “$\text{What}_N$” (what in the stage of Need) which is supposed to answer the questions like “How”, “How much”, “When”, and “Where”. Finally, at the Requirement phase, the aim is to form the set of “$\text{What}_R$” (what in the stage of Requirement) which can yield the final set of requirements.

Each stage in the figure uses the IPO (input-process-output) model. The process is formally defined as follows:
<table>
<thead>
<tr>
<th>Process</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
</table>
| Elicitation | $E_U \rightarrow W_E$ | - $E_U$ is the set of the Expectations of the User
- $W_E$ is the set of “What” obtained during the Elicitation process |
| Analysis | $W_E \rightarrow W_A$ | - $W_A$ is the set of “What” obtained during the Analysis process |
| Specification | $W_A \rightarrow W_S$ | - $W_S$ is the set of “What” obtained during the Specification process |
| V&V | $W_S \rightarrow W_{V&V}$ | - $W_{V&V}$ is the set of “What” obtained during the Verification & Validation process |
| $E_Ψ, E_N, E_R$: Elicitation | $E_Ψ$ is the first part of triple parts of Elicitation process done during the Wish phase
$E_N$ is the second part of triple parts of Elicitation process done during the Need phase
$E_R$ is the third part of triple parts of Elicitation process done during the Requirement phase |
| $A_Ψ, A_N, A_R$: Analysis | $A_Ψ$ is the first part of triple parts of Analysis process done during the Wish phase
$A_N$ is the second part of triple parts of Analysis process done during the Need phase
$A_R$ is the third part of triple parts of Analysis process done during the Requirement phase |
| $S_Ψ, S_N, S_R$: Specification | $S_Ψ$ is the first part of triple parts of Specification process done during the Wish phase
$S_N$ is the second part of triple parts of Specification process done during the Need phase
$S_R$ is the third part of triple parts of Specification process done during the Requirement phase |
| $V_Ψ, V_N, V_R$: V&V | $V_Ψ$ is the first part of triple parts of V&V process done during the Wish phase
$V_N$ is the second part of triple parts of V&V process done during the Need phase
$V_R$ is the third part of triple parts of V&V process done during the Requirement phase |
| $Phase_Ψ = E_Ψ; A_Ψ; S_Ψ; V_Ψ$ | The whole activities of the Whish phase obtain through composition of $E_Ψ, A_Ψ, S_Ψ,$ and $V_Ψ$ |
| $Phase_N = E_N; A_N; S_N; V_N$ | The whole activities of the Need phase obtain through composition of $E_N, A_N, S_N,$ and $V_N$ |
| $Phase_R = E_R; A_R; S_R; V_R$ | The whole activities of the Whish phase obtain |
through composition of $E_\mathcal{R}$, $A_\mathcal{R}$, $S_\mathcal{R}$, and $V_\mathcal{R}$

<table>
<thead>
<tr>
<th>Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Psi = {\omega_1, ..., \omega_n}$</td>
<td>The set of all wishes</td>
</tr>
<tr>
<td>$\mathcal{N} = {\eta_1, ..., \eta_m}$</td>
<td>The set of all Needs</td>
</tr>
<tr>
<td>$\mathcal{R} = {\tau_1, ..., \tau_p}$</td>
<td>The set of all Requirements</td>
</tr>
</tbody>
</table>

\[
\mathcal{N} = \{\forall \omega \in \Psi \mid Phase_\mathcal{N}(\omega)\}
\]

\[
\mathcal{R} = \{\forall \eta \in \mathcal{N} \mid Phase_\mathcal{R}(\eta)\}
\]

\[
\mathcal{R} \subseteq \mathcal{N} \subseteq \Psi
\]

Requirements management activities can be formally defined by using simple temporal logic as follows:

- $\Box (\neg \text{Finish (Development Process)} \Rightarrow \text{Do (Requirements Change Management)})$
- $\Box (\neg \text{Finish (Development Process)} \Rightarrow \text{Do (Requirements Tracing)})$
- $\Box (\neg \text{Finish (Development Process)} \Rightarrow \text{Do (Requirements Identification)})$

The reason for choosing temporal logic is that the management activities are in fact umbrella activities ongoing throughout the RE process.

---

5- Related Works

During the recent years, a few numbers of surveys have been conducted to review the current status of research in RE. Zave [70] draws a classification schema for research efforts in RE. The survey conducted by Davis [71] in 2005 and the latest version [72] shows the economical aspects of investment in RE research. Davis explores the macro-economic implications of RE research. The new version of paper [72] includes about 5200 publications and analysis of RE trends by especial focus on journals. Höfer in [73] provides an assessment of the status of empirical software research. The paper concentrates on empirical work in software engineering in ESE journal. In 2002 Glass et al. [74] introduced a classification scheme which differentiates papers in the field of computing based on five characteristics: topic, research approach, research method, reference discipline, and the level of analysis. Also, Roel [75] presents a paper classification and evaluation criteria for the RE conference.

The main difference between the current research and the previous ones is that it mostly emphasize on the RE research trends based on main and important activities in our definition. Other surveys use other viewpoints such as economy [71], but our viewpoint is based on activities in the RE process. By virtue of our viewpoint, we have reached an RE research tree. In our approach, the research trend has been recorded separately for each activity.

Another important difference is that the research resources of the existing surveys is based solely on publications but, the current research resources incorporates a wide diversity of publications, tools, researchers, projects, research groups, PhD theses, books, and scopes of journals and conferences. By defining a research method we haven’t gathered our research resources in a blind manner. In fact, the research
method has enabled us with a means for gathering more valuable resources among the whole existing research materials.

6. Research Method and Limitations

6.1. Research Method

In this section, we explain the research method used for identification, elicitation, and classification of the RE research records. As it is illustrated in Figure 5, the proposed research method is iterative and spiral. It has been started by a comprehensive study of the existing research resources and references in the field of RE with a special attention to the RE textbooks [10, 36, 76, 77, 78].

As illustrated in Figure 5, at the first iteration, we have drawn the initial plan and gathered the required and related resources. Also during the iteration, the initial knowledge about concepts and definitions related to requirements engineering have been obtained. The various types of gathered resources include creditable conferences, journals, papers, textbooks, PhD. dissertations, research and industrial groups and projects, clear-sighted researchers and professors, and offered courses in the field of requirements engineering. Although it may be possible to collect all the aforesaid resources, it is very time-consuming and unnecessary. Even in cases where there is a need to investigate the whole existing activities in the research scope, regardless of their quality, there exists techniques like random sampling which is based on the proven and precise mathematical theories of statistics and probability. By minimizing the statistical space, it will be possible with a close approximation to reality. In cases like the study of the research records conducted in a particular field, identifying valuable resources and references and not all the existing records in that field is important and noticeable.

Figure 5. The research method
As it can be seen in figure 6, an iterative and incremental approach has been used to collect highly-considered records and to avoid a blind search. Contrary to the linear and sequential approach in which each research resource is identified and collected separately and independently of other resources, in this approach different research resources are identified gradually and in parallel with attention to each other.

![Diagram of resource gathering method](image)

Figure 6. The resource gathering method

Making dependencies in identifying research records of diverse resources has led us from one credible resource to the other credible resource. For example, after identifying a credible journal in the RE field, by referring (recourse) to published papers in the previous issues, we add the related papers to the selected paper list. Furthermore, the editorial boards of the journals are eligible candidates to be added to the list of well-known researchers. In the same way using a nested approach, the collected information from one resource completes information of the other resources and the loop is repeated as long as new information is collected. In general, we have gathered the research resources and research activities for each resource with an iterative and evolutionary approach.

After extraction and identification of diverse research history in the field of RE, it is time for classification. For this aim, each of the mentioned records is studied and examined up to the extent which the appropriate class can be determined. For instance, as illustrated in figure 5, for papers it fluctuates from abstract review to the in-depth study of the body.

Finally the whole research scope in the field of RE contains diverse research activities which can be classified and entitled by the following topics: elicitation, analysis, specification, validation and verification, management, modeling, tracing, RE process, necessity, shortcomings and also service, object, and agent oriented RE. It is worth noting that the measure of creditability and quality of diverse resources are different but certain. For example, our measure to select a journal, alongside considering its relationship to RE, is standard measures like ISI and even their impact factor (IF), citation, and indexing considered as the last part of each cycle in the proposed research method entitled “Citation Analysis” as illustrated in figure 5. Also the selection measure for papers is their authors and publisher. Figure 7 represent the described research method from a new view point.
Figure 7. The proposed research method from a new viewpoint

Table 6 shows the number of each research resource gathered by the proposed method illustrated in Figure 6. The more information about the details of the research records of each resource has been prepared in a structured format and is publically accessible via [http://ceit.aut.ac.ir/islab/Researches/RE/Appendix.doc](http://ceit.aut.ac.ir/islab/Researches/RE/Appendix.doc).

For example, the details about PhD dissertation includes information like the title of dissertation, student and the related supervisor, the finish time, university, and the internet address to access the dissertation file (if any).

It is worth mentioning that currently there are more than 5000 papers published in the field of RE [36]. The research method described before has led us to 385 papers out the existing papers. We claimed before that using our research method can conduct us with high-quality records for each resource. The examination of the credit of the collected 385 papers is an evidence for the claim. The credit measure of a paper is its authors and the publisher.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools</td>
<td>72</td>
</tr>
<tr>
<td>Conferences</td>
<td>9</td>
</tr>
<tr>
<td>Journals</td>
<td>12</td>
</tr>
<tr>
<td>Papers</td>
<td>385</td>
</tr>
<tr>
<td>Researchers</td>
<td>81</td>
</tr>
<tr>
<td>Books</td>
<td>47</td>
</tr>
<tr>
<td>Projects</td>
<td>33</td>
</tr>
<tr>
<td>Department, Research Groups &amp; Centers</td>
<td>35</td>
</tr>
<tr>
<td>Courses</td>
<td>11</td>
</tr>
<tr>
<td>Thesis</td>
<td>37</td>
</tr>
<tr>
<td>Others</td>
<td>16</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>738</strong></td>
</tr>
</tbody>
</table>

As it can be seen in Table 6, it is evident that the number of papers is greater than the others. It is because the results of other research activities performed in other resources are usually reported in a paper format. It is worth mentioning that more than 5000 existing related papers were reviewed to select a set of 385 papers. The selection was mainly based on their authors and the publication location (journals and
conferences). In our opinion, papers can show the trends in the best way. We believe the 385 chosen papers do indeed constitute an excellent representation of research in the RE field.

6.2. Study Limitations

Limitations of this research fall into these categories:

a. The complexity of categorizing papers due to ambiguity or falling into several categories.

b. The poor quality of search engines available for use in finding primary sources, and the difficulty of getting consistent results from these and not having access to all resources.

7. Results

In this section we present the results of the study. The results are exhibited in two parts. In the first part the results exhibit the RE research trends and issues from the viewpoint of papers. The results reported in the second part show the RE research trends and issues from the viewpoint of tools, research groups, and projects, which we refer to them in the rest of this section as other resources in contrast with papers. In the both aforesaid parts, results are presented by both value and percentage.

7.1. Paper Share by important research topics

Table 7 shows paper share by important research topics in the field of requirements engineering during a 9-year period from 2002 to 2010. According to Table 7, specification has been the major research issue during the mentioned period on the average. Analysis occupies the second rank in this criterion. Elicitation, “others”, change management, V&V, modeling, goal-oriented RE, scenario-based RE, tracing, negotiation, and identification occupy the next places respectively from the viewpoint of mentioned criterion averagely. It is worth mentioning that the miscellaneous research activities done in the field of RE such as RE process, application, necessity, shortcomings and also service, object, and agent oriented RE are totally constitute a distinct research topic entitled “others”.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Year</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elicitation</td>
<td></td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td><strong>49</strong></td>
</tr>
<tr>
<td>Negotiation</td>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td><strong>10</strong></td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
<td>4</td>
<td>10</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>6</td>
<td>9</td>
<td>4</td>
<td><strong>58</strong></td>
</tr>
<tr>
<td>Specification</td>
<td></td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>10</td>
<td>12</td>
<td>11</td>
<td>13</td>
<td>6</td>
<td><strong>73</strong></td>
</tr>
<tr>
<td>Modeling</td>
<td></td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td><strong>29</strong></td>
</tr>
<tr>
<td>V&amp;V</td>
<td></td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td><strong>32</strong></td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Management</td>
<td></td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td><strong>34</strong></td>
</tr>
<tr>
<td>Tracing</td>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td><strong>13</strong></td>
</tr>
<tr>
<td>Identification</td>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td><strong>10</strong></td>
</tr>
</tbody>
</table>
As it can be seen in Figure 8, Among 385 papers, 19% of papers are about specifications, 15% are about analysis, 13% are about elicitation, 9% are about change management, 8% are about verification and validation, 7% are about modeling, 6% are about goal-oriented RE, 3% are about negotiation, scenario-based RE, tracing, and identification equally, and the rest (11%) are about other activities. As it was mentioned earlier in this survey, results show that specification has been the most important research issue in RE.

![Figure 8. Publication share by important research topics](image)

### 7.2. Other resources share by important research topics

As we said in the previous section, the results of the survey are presented in two categories. In section 7-1 results for publication were presented during the mentioned period; in this section other resources including tools, research groups, and projects are investigated. Similar to the previous section, the results are presented by both value and percentage.

Table 8 presents other resources share by important activities. Based on the results in this table, management is the major activity for tools, specification and V&V are
the major activity for research groups and ultimately, management and specification are the major activities for projects.

Table 8. Other resources share by important research topics

<table>
<thead>
<tr>
<th>Topic</th>
<th>Resource</th>
<th>Tools</th>
<th>Research Groups</th>
<th>Projects</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Development</td>
<td>Elicitation</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Negotiation</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Analysis</td>
<td>12</td>
<td>4</td>
<td>1</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Specification</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Modeling</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>V&amp;V</td>
<td>6</td>
<td>6</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>Requirements Change Management</td>
<td>Change Management</td>
<td>20</td>
<td>3</td>
<td>6</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Tracing</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Identification</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Approaches Goal-Oriented RE</td>
<td>Goal-Oriented RE</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Scenario-Based RE</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>72</td>
<td>35</td>
<td>33</td>
<td>140</td>
</tr>
</tbody>
</table>

The same results are shown in Figure 9. This figure corresponds to Table 8 and shows a schematic view of results in the table.
As visually depicted by Figure 10, the percentage of research in RE based on important activities are presented. Figure 10, shows that the other resources share by change management is 21%, specification is 16%, analysis is 12%, V&V is 11%, modeling is 9%, elicitation is 8%, tracing and negotiation are equally 7%, scenario-based and goal-oriented RE are equally 4% and finally identification is 1%.

![Figure 10. Other resources share by important research topics](image)

Finally we illustrate the total shares of publication and other resources by important research topics together in a single chart depicted in figure 11 during the mentioned 9-year period from 2002 to 2010.

![Figure 11. Publication and other resources share by important research topics](image)
7.3. RE Research Tree

Up to this point, we have described an analytical review of activities in the RE field. In this part, we present a research tree in the field of requirements engineering obtained as the final output of our research method explained in section 6-1. The tree, shown in Figure 12, illustrates in summary the major activities and research fields. Although there may be other fields, the fields depicted in the tree are the major ones. As it is shown in the tree, the major research areas in RE are important activities defined in section 3. It is worth noticing that to draw the tree in first phase major categories have been considered and in the second phase each existing category have been examined to derive some subcategories based on their corresponding share in the total. In fact, in each category, items which have received remarkable attention are considered as a subcategory. It is worth mentioning that Appendix A elaborate the details of the proposed RE research tree. It makes it possible to access all related resources including publications, tools, researchers, projects, research groups, PhD theses, and books.

8. Future work

In future we are going to deepen into each research issue to elaborate the current open issues in each field. Bearing such ultimate goal in mind, as the first step the research tree presented is revised and completed. Also, statistical methods can be used to improve the selection process and data analysis.

9. Summary and Conclusion

The primary intention in performing this research was to present our definition for requirements engineering and its process and activities. To support our definition, we have reviewed the research activities in the field of RE based on our classification scheme. For each activity in the process, the related research has been reviewed using an engineering approach. This paper does not claim that it covers all the research issues in the field, but it can be considered as a guideline for RE researchers to get to know the definitions and research activities in the RE field.

The findings out of the results analysis show that RE research is not confined to just main activities and other research fields are emerging. Also, the results show that specification and management have been the main important research activities in the field of RE.

The paper also presents an RE research tree. According to the proposed tree, in most of the main categories including analysis, V&V, specification, and management, formal methods is considered as a subcategory which illustrates the important role of the formalism in the RE research area.

Acknowledgments, This research has partially used financial support of ITRC (Iran Telecommunication Research Center). We are also grateful to Zoubida Kedad, and Peri Locoupolous for their comments and helps during the sabbatical period of Professor Barfroush at Loughborough university.
Figure 12. RE research tree
References


[65] Robertson S., Robertson J., Mastering the requirements process, Addison-Wesley, 1999.


APPENDIX A:

The details of the proposed RE research tree